AD-A245 344

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DEVELOPMENT OF THE DAMAGE CONTROL SYSTEMS ASSIST TOOL

by

William T. Carney
September 1991

Thesis Advisor: C.T. Wu
Second Reader: B.B. Giannotti

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/ AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DCLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL *(If Applicable)* 37	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS *(city, state, and ZIP code)* Monterey, CA 93943-5000	7b. ADDRESS *(city, state, and ZIP code)* Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 6b. OFFICE SYMBOL *(If Applicable)*	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS *(city, state, and ZIP code)*	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
DEVELOPMENT OF THE DAMAGE CONTROL SYSTEMS ASSIST TOOL

12. PERSONAL AUTHOR(S)
Carney, William T.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(year, month,day)*	15. PAGE COUNT
Master's Thesis	FROM TO	September 1991	51

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES

FIELD	GROUP	SUBGROUP

18. SUBJECT TERMS *(continue on reverse if necessary and identify by block number)*
Hypercard, Hypertext, Damage Control, Paperless Ship

19 Abstract (*continue on reverse if necessary and identify by block number*)

The paperless ship concept first proposed by VADM Metcalf has been advocated at the highest levels in the Navy. ARGOS is a prototype multi-media database interface system under development at the Naval Postgraduate School in support of this advanced concept. This thesis has implemented a shipboard systems evaluation and assistance tool called Systems Assist Module (SAM). SAM automates many of the evaluation and information retrieval processes that a Damage Control Assistant is required to perform during normal and emergency conditions. It demonstrates the superior capabilities attainable in a system implemented with economical, off-the-shelf technology.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT X UNCLASSIFIED/UNLIMITED __ SAME AS RPT. __ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL C.T. Wu	22b. TELEPHONE (Include Area Code) (408) 646-3391 22c. OFFICE SYMBOL CSGg

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted SECURITY CLASSIFICATION OF THIS PAGE

All other editions are obsolete Unclassified

Approved for public release; distribution is unlimited.

DEVELOPMENT OF THE DAMAGE CONTROL SYSTEMS ASSIST TOOL

by

William T. Carney
Lieutenant , United States Navy
B.S., Jacksonville University, 1984

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
SEPTEMBER 1991

Author: William T. Carney
William T. Carney

Approved by:
C.T. Wu, Thesis Advisor

B.B. Giannotti, Second Reader

for
David R. Whipple, Chairman, Department of Administrative Sciences

ABSTRACT

The paperless ship concept first proposed by VADM Metcalf has been advocated at the highest levels in the Navy. ARGOS is a prototype multi-media database interface system under development at the Naval Postgraduate School in support of this advanced concept. This thesis has implemented a shipboard systems evaluation and assistance tool called Systems Assist Module (SAM). SAM automates many of the evaluation and information retrieval processes that a Damage Control Assistant is required to perform during normal and emergency conditions. It demonstrates the superior capabilities attainable in a system implemented with economical, off-the-shelf technology.

TABLE OF CONTENTS

I. INTRODUCTION

The technology of todays armed services is immense. In fiscal year 1990, over ten percent of the Department of the Navy's budget was spent on the procurement of automated information systems, and associated equipment.[1] A high percentage of these systems comprise personnel administration and individual service functions, such as payroll accounting and database management. Relatively few involve critical information retrieval necessary for ongoing operations.

To keep a warship combat ready is the key to survival in our Navy. Yet with all of todays technological innovations and the Department of Defense commitment to using technology in order to maintain a decisive competitive advantage, only a limited amount of resources have been allocated to the automation of one of the most crucial aspects of shipboard survivability: damage control.

Currently, much of the data that is required in making a decision regarding shipboard damage control is either contained in publications (such as the ships Damage Control Book and the Ships Information Books) or can be found on damage control diagrams which must be manually assimilated. These diagrams, commonly referred to as "plates", measure 27 inches by 38 inches and are cumbersome to work with. Manual data retrieval methods typically increase the probability of selecting incorrect data, which in turn results in an erroneous evaluation. Not only is the validity of the information retrieved suspect , but also

[1] House Appropriations Committee, House of Representatives, 102nd Congress, "Department of Defense Appropriations Bill, 1992", Report 102-000, June,1991.

the resulting lack of timely information dissemination, caused by manual retrieval methods, may hamper shipboard survivability actions.

These problems can be solved with the development of an application that will greatly assist the Damage Control Assistant and personnel assigned to supervise each of the ships repair lockers in their duties by automating specific functions which must be accomplished. These would include, but are not limited to: searching various damage control diagrams for information pertaining to a given situation; arranging for the electrical and/or mechanical isolation of a specific space; ordering firemain segregation for the varying conditions of readiness [2] ; and assigning fire, smoke, flooding boundaries to a specific emergency condition. This application, Systems Assist Module (SAM), will be developed in accordance with the ARGOS architecture [Ref. 1]. The ARGOS architecture specifies a multi-media interface that provides access to multiple real time databases.

"Effective human-computer interaction relies on the [users] being able to develop an accurate mental model of the way a system functions."[3] The SAM application interface will support text, graphics, and audio to provide the user with maximum support. The goal is to achieve the most desirable interface possible, by way of maximizing user consistency and habit patterns while minimizing user effort and memorization.

The environment in which SAM was developed is Hypercard™, whose scripting language is Hypertalk™. Information on this environment can be found in Apples' Hypercard User's Guide [Ref. 2]. This environment proved to be quite

---

[2] The conditions of readiness are classified as "xray", "yoke", "zebra", "william", and "circle william". Each condition refers to a specific state of integrity and compartmentalization.

[3] Marshal, Nelson, and Gardiner, 1987, p. 229.

efficient in the production of the prototype application. SAM and Hypercard operate on Apple Macintosh computers[4]. An important point is that SAM can be ported for use on any Microsoft disk operating system (MSDOS) platforms which would facilitate its implementation throughout the fleet. This is accomplished with the use of Microsoft Windows Toolbook^{Tm} application.[5]

The purpose of this thesis is to demonstrate the viability of an automated systems assist tool that will aid in the timeliness of information dissemination during emergency operating conditions, as well as normal inport and underway conditions.

[4] Hypercard, Hypertalk, and Macintosh are trademarks of Apple Computer, Inc.
[5] Windows and Toolbook are trademarks of Microsoft Corporation.

II. PROBLEM STATEMENT

The need for an automated damage control information system is more clearly evident in this world of changing political alliances and techno-terrorists. Emergency situations may arise at any moment, in port or at sea, whether in condition I (wartime), or condition IV (peacetime). To combat these events, we rely on manually assimilating countless technical manuals and ships information books. Real-time information retrieval and dissemination in the area of damage control is practically nonexistent.

There has been research and development in the areas of stability and rapid communications. Naval Sea Systems Command is sponsoring a project to place shipboard maintenance and technical manuals onto nine track 6250 bits per inch, reel-to-reel tape (which can then be converted to other media). This is a step in the right direction. Unfortunately these computerized tapes never reach the level that could benefit from the automation - the unit or shipboard level. There is currently no effort to automate information retrieval for the purpose of making real-time evaluations in a survivability situation. Yet recent history has demonstrated this need. In 1987, the USS STARK (FFG-31) was struck by two Excocet missiles while operating in the Persian Gulf. Eighteen hours had passed before it was discovered that one of the two Excocet missiles had severed the firemain leading from the number one fire pump. Rated at 1000 gallons per minute, the pump put over one million gallons of sea water into the ship. The USS SAMUEL B. ROBERTS (FFG-58), after striking a mine while operating in the Persian Gulf, encountered massive progressive flooding. The drainage main bulkhead cutout located in auxiliary machinery room 2 had been damaged by the

blast, consequently the use of the eductor system to dewater, was not consudered to be an option. The decision-making process in these two instances could have been greatly enhanced with a graphically automated information system. In the case of the USS STARK (FFG-31), the drop in firemain pressure forward would have warranted an explanation, and the use of an automated system would have enabled the user to visualize what had occurred, thereby prompting the shut down of fire pump number one. As for the USS SAMUEL B. ROBERTS (FFG-58), in using such a system, it could could have been easily discerned that by properly aligning the drainage valves, it was possible to dewater using the eductor located in auxiliary machinery room number one. The initial actions, or lack thereof, in both cases were made by crewmembers whose emotions and adrenaline were peaked. Important aspects of damage control were overlooked by knowledgeable and experienced personnel. Assistance in the damage control decision making process is essential and is required immediately.

To become a damage control asset in the shipboard environment, the crewmember must not only be trained in firefighting and flooding containment, but also in the ability to retrieve pertinent information. An in depth knowledge of shipboard systems is required and the ability to search through a myriad of publications to find appropriate information is essential. The information is available onboard, but the time required to manually collect and evaluate the information can be vastly improved. There are very few fleet officers who can list all of the closures necessary to mechanically isolate a given space without first consulting a damage control book or diagram. This is truc because most of the Navy's engineering schools are are taught using generic ship classes or prototypes of engineering plants to train students. Upon completion of training,

it is then the responsibility of the individual crewmember, once assigned to his or her ship, to learn the ships specifics.

The Damage Control Assistant Course, located in Newport, Rhode Island, was created to specifically train individuals in all matters relating to damage control. The school maintains Damage Control Books and Diagrams for surface ships and trains each individual on the specifics of his or her ship. All personnel assigned the DCA billet must attend this school. The DCA, usually an Ensign or Lieutenant Junior grade assigned to a frigate or destroyer, a Lieutenant or higher rank on cruisers and larger vessels, is responsible, as per [Ref. 3]:

> Under the Engineer Officer, for establishing and maintaining effective damage control organization and for supervising repairs to hull and machinery. Specific duties [include]: (1) The prevention and control of damage; (2)Placing the ship in the condition of closure ordered by the Commanding Officer, ensure that appropriate closure classifications are assigned,...ensure that correct compartment check off lists are posted; (3) the operation, care, and maintenance of auxiliary machinery piping and drainage systems.

After only 8 weeks of training at the Damage Control Assistants Course and a subsequent week of live firefighting training, the DCA is the shipboard "expert" on damage control. Although the training received at both schools is demanding, it is not all inclusive. Experience gained through dedicated and persistent study is the only path to becoming an "expert" in damage control.

The need for an automated information system in the damage control area is essential. Knowledge of ships systems and assimilation of information regarding damage control practices only come with dedication and training which takes a substantial amount of time. The technology is available to overcome the shipboard damage control information bottleneck, as will be demonstrated by the SAM prototype.

III. IMPLEMENTATION

A. IMPLEMENTATION MACHINE AND LANGUAGE

The ARGOS system is being developed on Apple Macintosh micro-computers using the Hypercard programming environment and the Hypertalk scripting language. Hypercard is a programming environment which is based upon the interrelations of objects and events. The objects that exist within Hypercard are stacks, backgrounds, cards, containers, and buttons. Events refer to keyboard or mouse inputs. Examples of events are the mouse button being depressed, the cursor entering a defined space on the screen, or a command key combination. Any action that a user takes causes an event to be generated by the system. If the event is of interest to a particular application, such as depressing the mouse button, the system developer will have written a handler for that event. The handler defines, in the Hypertalk language, what actions the program will take. Handlers can be placed within any object and a message to invoke the handler is passed from one object to another, within a hierarchy, until the appropriate handler is found. The event handling feature of Hypertalk allows for a programmer to develop custom commands and functions that can be invoked from other objects within a stack. Hypercard provides a rich set of tools to support rapid prototyping and demonstration of concepts. By using Hypercard, it is possible to show the viability and feasibility of a concept in a short time. Hypercard is distributed by Apple as system software and comes free with every Macintosh computer.

The ARGOS system relies on digitized sound and high resolution graphics to effect its user interface. The Apple Macintosh micro-computer fully supports

these features without requiring the use of specialized software or added circuit boards.

B. ARGOS SYSTEM STRUCTURE

1. Modular Approach

The decomposition of a complex problem into many smaller, easily solved sub-problems results in a modular solution to the problem. There are benefits to this approach of problem solving. The sub-problems can be assigned to different programmers, with each solution being a self-contained module. As the individual modules are completed, they are integrated into a system that solves the initial complex problem. The total system can be developed incrementally. No one person must keep track of every detail of the implementation and many people may contribute to the solving of the large problem.

Modular systems are easy to upgrade. Because each module is a separate entity, upgrading the system is as simple as replacing individual modules. Modularity of stored data allows for the efficient use of permanent storage to hold databases. Frequently, modules must access common databases. Modularity also allows for expansion of system capabilities. New modules can be developed and added to the system with only minor changes to the existing system structure. The ARGOS system is currently composed of modules in the training, operations, maintenance, and administrative areas. Future modules can be added to expand the capabilities of the existing system.

To maintain the modular concept of ARGOS, SAM has been designed to operate using four separate stacks. These stacks hold data which represent similar information. Much like the relational database management systems

available today, Hypercard links the separate stacks to enable the user to retrieve information.[6] These stacks will be covered in depth later in this thesis.

2. Implementation

The logical functions that are inherent to the ARGOS system were identified and implemented in a manner supporting modularity. The operation of the system has three logical levels. The first level is the Selection level. In the Selection level, the user selects the ship containing the data of interest. The system identifies the user, through a password challenge, before allowing access to any ship's data or functions to work on the data. The next level is the Navigation level. The Navigation level describes the portion of the system where the user travels through the system and accesses the data that is to be worked on. This level may not exist in every module. The last level is the Functional Mode. The Functional Mode is the level where the work that the user intends to accomplish is done. This is the level at which you would find SAM.

The ARGOS system structure operates using a data construct called a stack. A stack is a last in, first out data structure and forms the structure for the ARGOS system. The stack structure offers an elegant solution for navigating through the system. As a user travels through the system, levels are pushed on the stack. The user can backtrack from any point within the system by simply pressing the "RETURN" button on the screen [Figure 1]. The action that the system takes is simply to pop the levels off the stack. The user perceives that he is tracing his path back to a prior level.

[6] Hypercard is considered by many to have relational capabilities, but is not a true relational database because Hypercard is not designed for generating reports, which is common for DBMS programs.

Figure 1. SAM Return button.

The stack structure meets the needs of the ARGOS system. It allows modular development of the total system and supports a networked computing environment.[7] It also allows a simple method of navigating through the system while fulfilling the user's semantic model of the system.

C. SAM DESIGN

1. Determine The Requirements

The first step taken in this automated systems design was to determine who the target user would be and examine the needs and requirements of that user group. To accomplish this, three different cross sections of personnel were queried: the 72 prospective Damage Control Assistants receiving training at the DCA School in Newport, RI; the 36 senior enlisted receiving training at Repair Locker Leader School at Treasure Island, CA; and the numerous officers attending Naval Postgraduate School who have previously held the DCA billet.

There were many very workable ideas from the three groups. These concepts ranged from automating inventories for damage control equipment, to developing a system to assist in the uncertainties of a Chemical, Biological, Radiological (CBR) environment. The most predominant themes which crossed these sections were: the need to gain faster response times in setting material conditions of readiness; the need to properly isolate spaces both mechanically and electrically; and the need for accessing information from DC books and

[7] This can be accomplished by placing the Hypercard application on a network server(via Appleshare or TOPS network).

diagrams in a timely manner. It is these three themes upon which this thesis is based.

2. Choosing the Model

For determining future applicability of the SAM prototype, this thesis focuses all module functions around a single shipboard space, the engine room of an FFG-7 class ship. The engine room was chosen to be modeled for two primary reasons, the first is its consistency with the ARGOS prototype structure. The second reason the engine room was modeled is because every system found in the damage control diagrams for the FFG-7 class has at least one component which either originates, terminates, or passes through the space. By modeling this one space, this thesis will demonstrate the viability of the Systems Assist Module.

3. Meeting the Users Requirements

The stacks created for SAM are the Core stack, the Damage Control Diagrams stack, Help stack, and the Closure Listing stack.

a. *The Core Stack*

To maintain compatibility with the ARGOS structure, all icons developed in the initial ARGOS project will be maintained throughout the Systems Assist Module. The user will open the module to the line schematic of an FFG, a copy of the first card of the functional level in ARGOS [Figure 2].

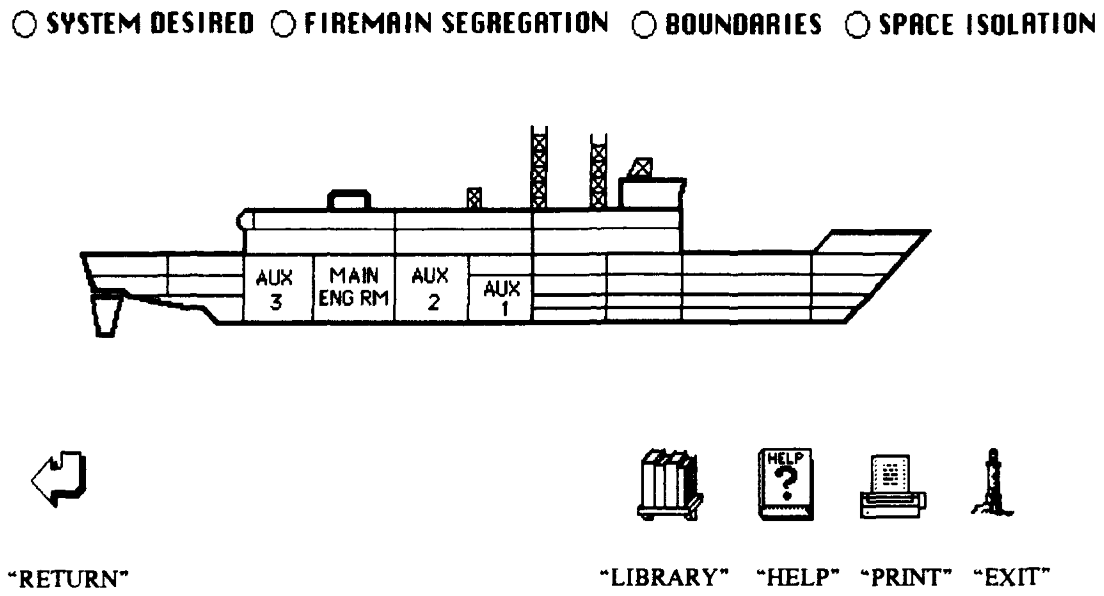

Figure 2. The System Assist Module "Base" card.

This card is the "base card" of the core stack within the Systems Assist Module. It is from this card that a user will choose which information is desired via buttons. Buttons are hot spots, either invisible or visible, which link related information through the Hypercard hierarchy. By simply clicking on the "SPACE ISOLATION" button, for example, Hypercard opens the card which "SPACE ISOLATION" relates. After the information desired is retrieved, the user will then click the "RETURN" button to return to the base card .

In order not to confuse the user with an excessive amount of buttons for each card, this thesis utilizes an external command (XCMD) to invoke pop-up menus. For example, to use individual icons or text to represent each of the 12 systems represented in the SAM stack would crowd the card and possibly distract the user in times of expediency. With the use of a pop-up menu the user simply clicks on the button "SYSTEM DESIRED" and, while continuing to hold

the mouse down, chooses between the systems which have become visible [Figure 3].

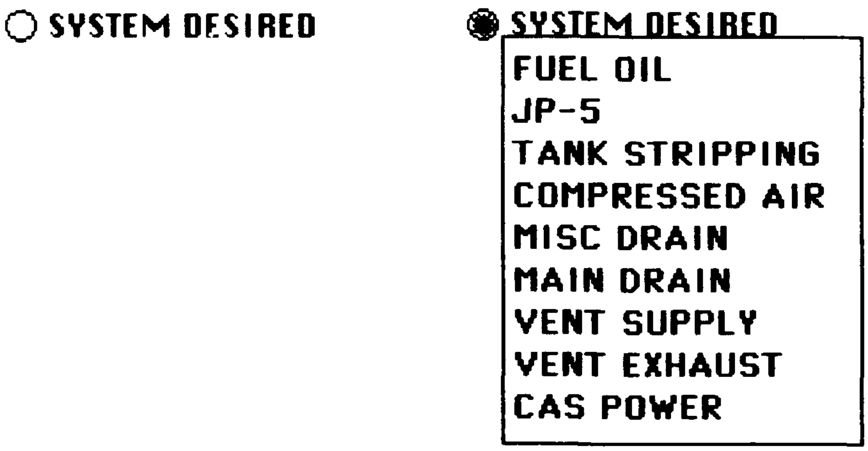

Figure 3. Example of SAM pop-up menu.

There are four buttons on the base card which have the pop-up feature: the "SYSTEMS DESIRED" button, the "FIREMAIN SEGREGATION" button, the "SET BOUNDARIES" button, and the "SPACE ISOLATION" button. The pop-up feature is an example of a resource. A resource can be a font, an icon, a sound, a picture, a palette, an external command, or an external function. This resource is an example of an external command and is broken into two components: a data fork and a resource fork. The data fork generally contains that information which we enter into an application. In this instance, when invoking the external command for creating a pop-up menu, the Hypercard programmer will write into a script "get pop-up menu". When this command is executed, Hypercard searches its hierarchy and will find the data forks complement, the resource fork. The resource fork is code which is other than user entered data, usually created and

compiled from higher level languages such as C or Pascal. The purpose in using resources comparable to the pop-up XCMD is twofold. First, the ease in which this reusable code was imported into SAM supported the ARGOS prototyping design criteria. Second, the concept of separating certain elements from the main program code offers simplicity to the application. A simple modification can be made quickly, instead of searching through hundreds of pages of code and recompiling the program.

Also located within the core stack is an example of how SAM could be used in times of emergency with the use of the "Boundaries" button. This button links the base card with cards that have a three-dimensional drawings of the engine room and surrounding spaces. Clearly marked on these cards are the primary and secondary fire, flooding, and smoke boundaries which would be ordered by the DCA to the repair locker leader. The repair locker leader would then supervise the setting and maintenance of these boundaries until ordered to secure. Having the boundary information instantly at the click of a button is clearly an advantage over manually searching the damage control diagrams for the information.

The remainder of the buttons located on the base card and throughout the SAM application (the return button, the library button, the help button, the print button, and the exit button) are consistent with the original ARGOS prototype.

b. ***The Damage Control Diagrams Stack***

In order to meet the user requirement of accessing information in a timely manner, this thesis postulated two possible solutions. The first of which was to use one of the many, inexpensive off-the-shelf draw applications to design an engine room graphic, and then create detailed systems graphics to

overlay the engine room graphic. It was determined, through actual design attempts, that creating such detailed graphics would be too time intensive and deter from the ARGOS theme of "rapid prototyping". The second solution, to use existing graphics and import them into the application was chosen.

All United States naval warships receive a set of damage control diagrams upon commissioning. These diagrams are created, by the ships commissioning activity, in a UNIX operating environment using a computer aided design application called CAD Forax. The data is then stored on 9 track, 6250 bits per inch, reel-to-reel magnetic tape, in order for the diagrams to be routinely maintained and updated. This computerization of existing information is the reason for choosing the second alternative.

The format in which the damage control diagrams are being developed is currently incompatible with the Apple hardware used in the ARGOS prototype. However, in order to fulfill the ship system information retrieval requirement, the System Diagram section of SAM has been accomplished by optically scanning FFG-7 class damage control diagrams (supplied by Long Beach Naval Shipyard). Once scanned, the diagrams are "cleaned" (removal of extraneous bits) using the paint application included with the scanner software.

The scanning was initially accomplished using a Hewlett Packard Deskjet Plus™ greyscale flatbed scanner.[8] Several test scans of diagrams were taken using the formats of line art, greyscale, and halftones, while changing picture resolution from 72 dots per inch to 300 dots per inch. The line art option at 300 dots per inch resolution was of the best quality among the test scans, and was initially chosen to be the scan of choice for the SAM application.

[8] Deskjet Plus is a trademark of Hewlett Packard, Inc.

The test scans were then exported to a Hypercard test stack to determine the horizontal and vertical coordinates for the scans. This is necessary in order to provide uniformity of all pictures imported into the Hypercard stack. Once exported, however, it was revealed that the Hypercard program allows only for the import of MacPaint or PICT file formats having a maximum resolution of 72 dots per inch. The Hypercard program automatically converted the smaller size test scans (500 kb or less) to 72 dots per inch, and rejected the larger scans. Five of seven test scans taken were over 500 kilobytes, and the scans which the program accepted were of such poor quality that system specifics, such as valve numbers and vital equipment alternate distribution sites, could not be discerned.

While Hypercard itself does not support color or greyscale images directly on its cards, it is possible to display such images in a window that overlays the current Hypercard window. By displaying color images in a window smaller than the card window, you can give the user the impression that the card contains color. This is accomplished using the Hypertalk picture command. There is a limited set of properties that apply to windows created with the picture command. They are rect, globalrect, loc, globalloc, scroll, zoom, scale, visible, and dithering. An example of how these properties are coded in Hypertalk is provided from one of SAM's' scripts [Figure 4].

```
on openCard
  lock screen
  picture "JP-5", file, window, false
  set the globalrect of window "JP-5" to "32,32,630,430"
  set the scroll of window "JP-5" to "100,50"
  set the visible of window "JP-5" to true
  unlock screen
end openCard
```

Figure 4. An example of Hypertalk's PICTURE command.

All coding from the SAM prototype can be found in the Appendix.

Images may be displayed from a number of different sources, including PICT or MacPaint files, as well as PICT resources available in stacks along the message hierarchy. Color will greatly enhance the user interface. Therefore, diagram scans were accomplished using the Micotek 600ZS Scanmaker™ scanner. [9] The resulting pictures from these scans are converted into the PICT format which can be read by Hypercard. All pictures have been cut to show only that information pertaining to engine rooms systems, and were not intended to be ship-inclusive.

When the image windows are open, Hypercard normally maintains the picture in memory, putting abnormal requirements on the computers random access memory (RAM). Due to these memory constraints, the damage control diagrams were scanned in at 120 dots per inch resolution. The trade off in quality can be minimized with an increase in RAM. For example, a 120 dots per inch color scan of the compressed air system uses 1.4 megabytes of memory, whereas the same system scanned at 300 dots per inch would consume a massive 8.6 megabytes of memory.

[9] Scanmaker is the trademark of Microteck, Inc.

There is also the speed of the central processing unit to consider. The increase in bytes of a file relates to the proportionate number of hard disk hits which must be made to place the image in memory. Therefore, the larger the image in bytes, the longer the time until the image is displayed. This timelag can be overcome with the use of accelerator cards, a faster hard drive, a faster central processing unit, or any combination of these.

c. *The Space Isolation Stack*

Ships are subdivided into as many small compartments as possible so that flooding can be restricted to a minimum. In order to make the subdivision effective, all boundaries between watertight compartments including the hull structure, doors, hatches, pipe and cable stuffing tubes, and ventilation closures must be maintained absolutely intact. Failure to maintain strict watertight integrity will lead to progressive flooding, which as the term implies means flooding from one compartment to another. This flooding, if not checked, will result in the eventual flooding of enough compartments to cause the ship to capsize or sink.

The only sure way to guard against progressive flooding is to rigorously observe the requirements for periodic watertight integrity tests and inspection required by Chapter 9881 of the NAVSHIPS Technical Manual [Ref. 5]. As each vessel is commissioned, all boundaries were tight to the extent designed and flooding would not extend beyond any boundary that was not actually damaged. To maintain this condition is the responsibility of the ship's force.

The tactics of modern naval warfare are such that little warning exists for attack by mine, bomb, missile or torpedo. These may strike a few seconds after a plane is sighted, torpedo wake is seen, or without any warning at all.

Typically, there is not an opportunity to close doors, hatches, or other fittings to prevent progressive flooding. For this reason, during all wartime cruising (condition I), all fittings should be kept secured except those which must absolutely be kept open to operate the ship or make it habitable.

Material condition checkoff lists should be posted in every compartment. These checkoff lists enumerate every fitting within that compartment, together with the damage control classification of that fitting and the division which is responsible for setting the fitting under various material conditions. The information from which the checkoff lists are created is found in the ships Damage Control Book [Ref. 5].

The Space Isolation stack was designed to attain quick response times in the event of damage. The stack is logically divided into two areas: electrical isolation and mechanical isolation. Each area gives the user the necessary information required to isolate the modeled space. By linking the areas via buttons, the user can directly open either area without having to return to the SAM base card. The information found in the Space Isolation stack is contained in fields. Hypercard fields are rectangular areas on a card where the user can type and edit easily. A scrolling field option was chosen because the amount of input information was greater than a normal card field could accommodate. A scrolling field, under Hypercard 2.0, has a limit of 30,000 characters. Having all needed information located in one field makes it possible to search for desired information easily. For example: damage has caused the sea chest cut out valve to be inoperable. The user can not recall what type and size the valve is in order to effect repairs. Instead of manually searching a damage control book, the user can simply click the mouse on the "Space Isolation" button on the base card of the core stack [Figure 5].

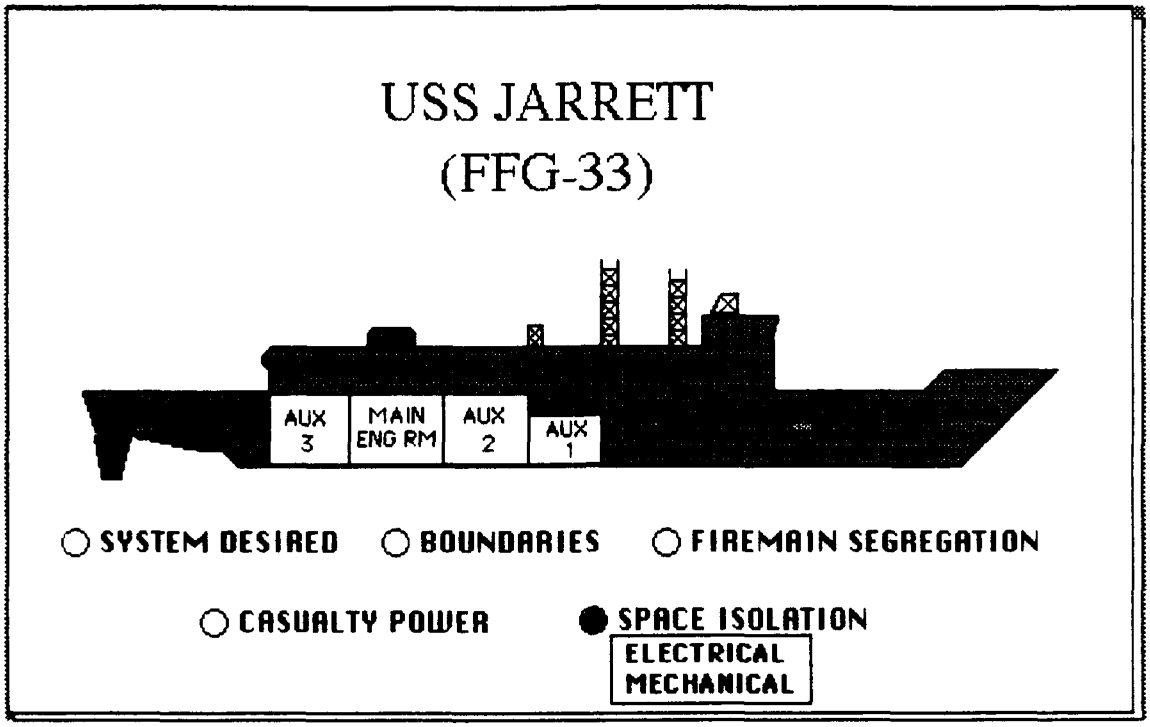

Figure 5. An example of the Space Isolation button being depressed.

The user then chooses "Mechanical Isolation" from the pop-up menu, and enters into the mechanical isolation area of the Space Isolation stack. Once in this area, the user clicks the mouse over the binoculars, which is the icon representing the search function [Figure 5]. To find the information requested the user is prompted to enter a search string. "Sea chest" is entered, the search function is performed on the field, and the desired information is then marked by a rectangle around the data. Not only can this function be used to retrieve data, but also can find similar data within the field.

Figure 6. Binocular Icon for the search function

When the user discovers during the search that the sea chest cut out is a 2 inch gate valve, he or she can enter a carriage return to determine if there are any other sea chest valves which fit this description. The user may also wish to search the field for any nonessential 2 inch gate valve in order to replace the inoperable sea chest cut out valve. In this instance he or she can once again click the binocular icon and enter "2 in gate valve" to see all 2 inch gate valves located within the field.

The search function used in this section of SAM is similar to the code written for the original ARGOS prototype, only the stack names were altered. This function is an extremely important feature which will not only give way to rapid response in emergency situations, but also access to valuable data in times of routine maintenance and repair.

d. *The Help Stack*

The move toward personal computers and workstations draws much of its impetus from two developments: a general increase in computer knowledge by the public and the desire by the end user to control their own computing. According to Elias Awad, however, this only comes about when the end user accepts the presence of an information center for end user training and technical service. [Ref. 6] This concept of reliance on the information center is outdated. The logarithmic increase in micrcomputing power seen in the past decade has given rise to the more decentralized concepts of application development and

use. Rapid prototyping is a classic case which downplays the role of the centralized information center.

In the Systems Assist Module, as in all prototypes based on the ARGOS architecture, the user will find a module specific Help function [Figure 7].

Figure 7. Help icon for Help facility.

This function replaces the need for the user to seek advice from an information center concerning basic operation of the system and training support. The intuitive graphical user interface allows the user to visualize what he or she is actually attempting to accomplish. The Help stack, like all other stacks in SAM, is equipped with the standard ARGOS navigational buttons. The stack itself is comprised of numerous fields with information on basic system operation, system navigation, button functions and descriptions. directions in performing searches and, specifically for the Damage Control Diagrams stack, an alphabetical listing (including graphical symbols) of all system components and their description. The last point is crucial to those users who are not yet familiar with the standard symbols relating to specific components of the systems diagrams.

Not only can this stack be used as a standard help facility, but it may also serve as a training module for those who have yet to master the damage control diagrams and those who are unfamiliar with ships systems. By opening the DC Diagrams stack via the "System Desired" pop-up menu, the user can quiz himself or herself on information found in the diagrams. The user can check if the

correct response was chosen by clicking the "Help" icon and performing a search on the component number. The user can get the same results by entering the "Space Isolation" stack and performing the same search, although the "Space Isolation" stack does not show component graphics.

The intuitive graphical user interface found in the Macintosh environment coupled with specific interactive help facilities used in SAM, promote a user-friendly computing environment from which the unit or ship can only benefit.

IV. CONCLUSIONS

The design and implementation of the System Assist Module for ARGOS has demonstrated the feasibility and advanced capabilities of the ARGOS development environment. It was constructed as an integrated data retrieval system based on modular design, which allows expansion with minimal revisions to the overall system. It uses a data structuring scheme that allows the modeling of physical objects; this results in a system that is more intuitive to the user and a closer representation of the user's environment.

The storage media most likely to be of greatest benefit for the System Assist Module is compact disk read only memory optical storage (CD ROM). This would be best for a number of reasons. The CD ROM disks will hold about 550 megabytes of data. This large volume is necessary to hold all of the data that is required to comprise a full set of damage control diagrams for a ship. When modifications occur to a ships systems, the modification(s) can be temporarily stored on a computers hard disk, or floppy disk. When a ship enters a routinely scheduled maintenance period, instead of receiving a set of modified damage control plates, the ship will receive an updated version of a CD ROM specific to the ship. The only change to the current computer aided design system will be the storage of data on CD ROM vice 6250 bpi magnetic tape. This change will not only allow the ship or unit the chance to automate damage control functions, but will also severely reduce physical storage space now required by the larger magnetic tapes held by shipyards and repair facilities. Optical disks are also quite sturdy and are not easily damaged by normal day to day use. This will provide

protection of the data as it is in transit to the user facility. Today the only true alternative for this amount of data is CD ROM.

Future areas of study directly relating to the Systems Assist Module application include:

- Linking SAM with the ARGOS prototype ordering system to enable rapid repair or replacement of damaged system components.
- Converting the current storage media of damage control plates from magnetic tape to CD ROM.
- Using external devices, such as temperature or flooding alarms to trigger a controlling action from the Systems Assist Module.
- Determination of which storage media is truly best for this application to include CD ROM, tape drives and removable hard disks.

This development of the System Assist Module supports the ARGOS environment in pursuit of the paperless ship concept. The ARGOS SAM module surpasses the functional requirements for current manual retrieval methods, and provides a superior user interface that will improve productivity and drastically reduce the process of information collection and dissemination in the shipboard damage control environment.

APPENDIX

SYSTEM ASSIST MODULE CODING

ALL THE SCRIPTS FROM STACK: CHIL 2:Applications:Hypercard:sam opener

```
------------------------------------------------
••••• STACK SCRIPT
on openStack
-- presents to the user an opening screen
  repeat 5 times
  play "nu age bass"
 end repeat
end openStack


••••• BACKGROUND No.1 Art


••••• BACKGROUND No.1  FIELD No.1 Category
on closeField
 set name of this card to me
end closeField


••••• BACKGROUND No.1  FIELD No.2 Name


••••• BACKGROUND No.1  FIELD No.3 Card Num


••••• CARD No.1 card 1
on openCard
-- gives anninmation to the opening scene of SAM
 show 19 cards
 go card "open" of stack "sam1"
end openCard
```

```
-- on opening of card 20, the show command is halted, then
-- the base card is opened
on openCard
 go next card
end openCard
```

••••• CARD No.21 card id 4800

```
on openCard
 wait 1 seconds
 visual effect iris open very slow
 go next card
end openCard
```

ALL THE SCRIPTS FROM STACK: CHIL 2:Applications:Hypercard:SAM1

••••• STACK SCRIPT

••••• BACKGROUND No.1 bkgnd id 2687

••••• BACKGROUND No.1 FIELD No.1 BOUNDARIES

••••• BACKGROUND No.1 BUTTON No.1 LIBRARY

```
on mouseUp
 PLAY "LIBRARY"
 push card
 go to card library OF STACK "SAM1"
end mouseUp
```

••••• BACKGROUND No.1 BUTTON No.2 HELP

```
on mouseUp
 PLAY "HELP"
 push this card
 go to stack "ARGOS HELP"
end mouseUp
```

```
••••• BACKGROUND No.1  BUTTON No.3  PRINT
on mouseUp
 play "PRINT"
 doMenu Print Card
end mouseUp


••••• BACKGROUND No.1  BUTTON No.4  EXIT
on mouseUp
 play "bye"
 go home
end mouseUp


••••• BACKGROUND No.1  BUTTON No.5  New Button
on mouseUp
 -- goes back to ship side view level
 visual effect zoom out
 go to card "BASE" of stack "sam1"
end mouseUp


••••• BACKGROUND No.1  BUTTON No.6  New Button
on mouseUp
 -- goes back to ship side view level
 visual effect zoom out
 go to card "BASE" of stack "sam1"
end mouseUp


••••• BACKGROUND No.2 Art


••••• BACKGROUND No.2  FIELD No.1 Category
on closeField
 set name of this card to me
end closeField


••••• BACKGROUND No.2  FIELD No.2 Name


••••• BACKGROUND No.2  FIELD No.3 Card Num
```

```
••••• CARD No.1 card id 19887
on openCard
 go next card
end openCard


••••• CARD No.2 open
on openCard
 wait 1 seconds
 visual effect iris open very slow
 go next card
end openCard


••••• CARD No.3 base
on openCard
 play "make a selection"
end openCard

on mouseupinpicture wname, cloc
-- automatically closes any window which may be open from
-- last use
 close window wname
end mouseupinpicture


••••• CARD No.3  FIELD No.1  Pop-up NoteFld 30
on enterInField
-- hide the pop-up field XCMD execution
 hide me
 send "hideShowPopUp" to card button id 30
end enterInField


••••• CARD No.3  BUTTON No.1  EXIT
on mouseUp
 play "bye"
 go home
end mouseUp


••••• CARD No.3  BUTTON No.2  PRINT
on mouseUp
 play "PRINT"
 doMenu Print Card
end mouseUp
```

```
••••• CARD No.3  BUTTON No.3  HELP
on mouseUp
 PLAY "HELP"
 push this card
 go to stack "ARGOS HELP"
end mouseUp


••••• CARD No.3  BUTTON No.4  LIBRARY
on mouseUp
 PLAY "LIBRARY"
 push card
 go to card "library" OF STACK "SAM1"
end mouseUp


••••• CARD No.3  BUTTON No.5  er
on mouseUp
-- displays the engine room schematic
 PLAY "ENGINE ROOM"
 lock screen
 picture "plate 2a",file, zoom, false
 set globalrect of window "plate 2a" to "32,35,567,383"
 set the scroll of window "plate 2a" to"50,100"
 set the visible of window "plate 2a" to true
 unlock screen
end mouseUp


••••• CARD No.3  BUTTON No.6  SYSTEM DESIRED
on mouseDown
-- creates the po-up menu for system desired button
 PLAY "SYSTEM DESIRED"
 put "FUEL OIL" into systems
 put return & "JP-5" after systems
 put return & "TANK STRIPPING" after systems
 put return & "COMPRESSED AIR" after systems
 put return & "VENTILATION EXHAUST" after systems
 put return & "VENTILATION SUPPLY" after systems
 put return & "CASUALTY POWER" after systems
 put return & "MAIN DRAINAGE" after systems
 put return & "MISCELANEOUS DRAINS" after systems
 get popupmenu(systems,0,0,0)
 if it <> 0 then
  put item 1 of it into row
```

••••• CARD No.3 BUTTON No.6 SYSTEM DESIRED (Cont.)

```
 if (row = 1) then
   push card
   go to card 1 of stack "dc diagrams"
  else if (row = 2) then
   push card
   go to card 2 of stack "dc diagrams"
  else if (row = 3) then
   push card
   go to card 3 of stack "dc diagrams"
  else if (row = 4) then
   push card
   go to card 4 of stack "dc diagrams"
  else if (row = 5) then
   push card
   go to card 5 of stack "dc diagrams"
  else if (row = 6) then
   push card
   go to card 6 of stack "dc diagrams"
  else if (row = 7) then
   push card
   go to card 7 of stack "dc diagrams"
  else if (row = 8) then
   push card
   go to card 8 of stack "dc diagrams"
  else if (row = 9) then
   push card
   go to card 9 of stack "dc diagrams"
  end if
 end if
end mousedown
```

••••• CARD No.3 BUTTON No.7 FIREMAIN SEGREGATION

```
on mouseDown
 play "firemain seg"
 put "XRAY AND YOKE" INTO FIREMAIN
 put return & "ZEBRA" after firemain
 get popupmenu(firemain)
 if it <> 0 then
  put item 1 of it into row
  if (row = 1) then
   push card
   go to card 10 of stack "dc diagrams"
  else if (row = 2) then
   push card
```

••••• CARD No.3 BUTTON No.7 FIREMAIN SEGREGATION (Cont.)

```
    go to card 11 of stack "dc diagrams"
   end if
  end if
end mousedown
```

••••• CARD No.3 BUTTON No.8 BOUNDARIES

```
on mouseDown
  put "FIRE BOUNDARIES,FLOODING BOUNDARIES,SMOKE
BOUNDARIES" ¬
  into boundaries
  get popupmenu(boundaries)
  if it <> 0 then
   put item 1 of it into row
   if (row = 1) then
    push card
    go to card "FIRE BOUNDARIES" of this stack
   else if (row=2) then
    push card
    go to card "flooding boundaries" of this stack
   else if (row=3) then
    push card
    go to card "smoke boundaries" of this stack
   end if
  end if
end mousedown
```

••••• CARD No.3 BUTTON No.9 SPACE ISOLATION

```
on mouseDown
  play "space isolation"
  put "ELECTRICAL,MECHANICAL" into isolation
  get popupmenu(isolation)
  if it <> 0 then
   put item 1 of it into row
   if (row = 1) then
    push card
    go to card 1 of stack "closure"
   else
    go to card 2 of stack "closure"
   end if
  end if
end mousedown
```

```
••••• CARD No.3  BUTTON No.10  CASUALTY POWER
on mouseUp
 go to card "casualty power"
end mouseUp


••••• CARD No.3  BUTTON No.11  sorry
on mouseUp
 play "sorry this area has not"
end mouseUp


••••• CARD No.3  BUTTON No.12  sorry
on mouseUp
 play "sorry this area has not"
end mouseUp


••••• CARD No.3  BUTTON No.13  sorry
on mouseUp
 play "sorry this area has not"
end mouseUp


••••• CARD No.4 fire boundaries
on mouseUpInPicture
 close window "plate 2a"
end mouseUpInPicture


••••• CARD No.4  BUTTON No.1  return
on mouseUp
 play "return"
 visual effect wipe left slow
 go to card "base" of stack "sam1"
end mouseUp


••••• CARD No.4  BUTTON No.2  New Button
on mouseUp
 -- goes back to ship side view level
 visual effect zoom out
 go to card "BASE" of stack "sam1"
end mouseUp
```

```
••••• CARD No.4  BUTTON No.3  er
on mouseUp
 PLAY "ENGINE ROOM"
 lock screen
 picture "plate 2a",file, zoom, false
 set globalrect of window "plate 2a" to "32,35,567,383"
 set the scroll of window "plate 2a" to"50,100"
 set the visible of window "plate 2a" to true
 unlock screen
end mouseUp


••••• CARD No.5 casualty power


••••• CARD No.5  FIELD No.1  card field id 1


••••• CARD No.5  BUTTON No.1  return
on mouseUp
 play "return"
 visual effect wipe left slow
 go to card "base" of stack "sam1"
end mouseUp


••••• CARD No.5  BUTTON No.2  CASUALTY POWER DIAGRAM
on mouseUp
 GO TO  CARD "casualty power" of stack "dc diagrams"
end mouseUp


ALL THE SCRIPTS FROM STACK: CHIL 2:Applications:Hypercard:closure
---------------------------------------------------
••••• STACK SCRIPT


••••• BACKGROUND No.1 bkgnd id 2606


••••• BACKGROUND No.1  FIELD No.1 Description


••••• BACKGROUND No.1  FIELD No.2 TYPE
```

```
••••• BACKGROUND No.1  BUTTON No.1  LIBRARY
on mouseUp
 PLAY "LIBRARY"
 push card
 go to card library OF STACK "SAM1"
end mouseUp


••••• BACKGROUND No.1  BUTTON No.2  HELP
on mouseUp
 PLAY "HELP"
 push this card
 go to stack "ARGOS HELP"
end mouseUp


••••• BACKGROUND No.1  BUTTON No.3  PRINT
on mouseUp
 play "PRINT"
 Print bg field "description"
end mouseUp


••••• BACKGROUND No.1  BUTTON No.4  EXIT
on mouseUp
 play "bye"
 go home
end mouseUp


••••• BACKGROUND No.1  BUTTON No.5  Find
on mouseUp
 -- this handler provides for a modified search.
 put the id of this card into tempid
 PLAY "SEARCH"
 ask"Please enter Search String."
 if visible of background field "Description" then
  set lockscreen to true
  put "find string" && quote & it & quote && "in field Description"¬
  into msg
  hide msg
  send returnkey to hypercard
  if tempid <> id of this card then
   go recent
   set lockscreen to false
  end if
```

```
••••• BACKGROUND No.1  BUTTON No.5  Find (Cont.)
  else
    hide msg
    put "find string" && quote & it & quote && "in field type" into msg
    hide msg
    send returnkey to hypercard
  end if
end mouseUp


••••• BACKGROUND No.1  BUTTON No.6  return
on mouseUp
  play "return"
  visual effect wipe left slow
  go to card "base" of stack "sam1"
end mouseUp


••••• BACKGROUND No.1  BUTTON No.7  New Button
on mouseUp
  -- goes back to ship side view level
  visual effect zoom out
  go to card "BASE" of stack "sam1"
end mouseUp


••••• CARD No.1 electrical isolation


••••• CARD No.1  BUTTON No.1  Mechanical Isolation
on mouseUp
  push card
  go to card "mech isolation"
end mouseUp


••••• CARD No.2 mech isolation


••••• CARD No.2  BUTTON No.1  Electrical Isolation
on mouseUp
  push card
  go to card "electrical isolation"
end mouseUp
```

ALL THE SCRIPTS FROM STACK: CHIL 2:Applications:Hypercard:DC DIAGRAMS

```
••••• STACK SCRIPT
on mouseDownInPicture
 -- ensures all picture windows are closed
  close window wname
end mouseDownInPicture


••••• BACKGROUND No.1 bkgnd id 2630


••••• BACKGROUND No.1  BUTTON No.1  return
on mouseUp
-- closes specific picture windows which may have been left open
-- whileswitching stacks
 close window "fm seg x & Y"
 close window "fm seg z"
 close window "cas power"
 play "return"
 visual effect scroll left
 go to card "base" of stack "sam1"
end mouseUp

on mouseUpInPicture
 -- ensures all windows are closued prior to leaving module
 close window "fm seg x & Y"
 close window "fm seg z"
 close window "cas power"
 close window "comp air"
 close window "fuel oil"
 close window "jp-5"
 close window "vent supply"
 close window "vent exhaust"
 close window "misc drains"
 close window "main drain"
end mouseUpInPicture
```

```
••••• BACKGROUND No.1  BUTTON No.2  LIBRARY
on mouseUp
 PLAY "LIBRARY"
 push card
 go to card library OF STACK "SAM1"
end mouseUp


••••• BACKGROUND No.1  BUTTON No.3  HELP
on mouseUp
 PLAY "HELP"
 push this card
 go to stack "ARGOS HELP"
end mouseUp


••••• BACKGROUND No.1  BUTTON No.4  PRINT
on mouseUp
 play "PRINT"
 doMenu Print Card
end mouseUp


••••• BACKGROUND No.1  BUTTON No.5  EXIT
on mouseUp
 play "bye"
 go home
end mouseUp


••••• BACKGROUND No.1  BUTTON No.6  Mechanical Isolation
on mouseUp
 go to card "mech isolation" of stack "closure"
end mouseUp


••••• BACKGROUND No.1  BUTTON No.7  to base
on mouseUp
 -- goes back to ship side view level
 visual effect zoom out
 go to card "BASE" of stack "sam1"
end mouseUp
```

```
-- the following card scripts open the defined diagram in a window
-- which overlays Hypercard.

••••• CARD No.1 fuel oil
on openCard
 lock screen
 picture "Fuel Oil", file, window, false
 set the globalrect of window "Fuel Oil" to "32,32,630,430"
 set the scroll of window "Fuel Oil" to "100,50"
 set the visible of window "Fuel Oil" to true
 unlock screen
end openCard

on mouseUpInPicture
 close window "fuel oil"
end mouseUpInPicture


••••• CARD No.2 JP-5
on openCard
 lock screen
 picture "JP-5", file, window, false
 set the globalrect of window "JP-5" to "32,32,630,430"
 set the scroll of window "JP-5" to "100,50"
 set the visible of window "JP-5" to true
 unlock screen
end openCard

on mouseUpInPicture
 close window "jp-5"
end mouseUpInPicture


••••• CARD No.3 Tank Stripping
on openCard
 lock screen
 picture "Stripping", file, window, false
 set the globalrect of window "Stripping" to "32,32,630,430"
 set the scroll of window "Stripping" to "100,50"
 set the visible of window "Stripping" to true
 unlock screen
end openCard

on mouseUpInPicture
 close window "stripping"
end mouseUpInPicture
```

```
••••• CARD No.4 Compressed Air
on openCard
 lock screen
 picture "Comp Air", file, window, false
 set the globalrect of window "Comp Air" to "32,32,630,430"
 set the scroll of window "Comp Air" to "100,50"
 set the visible of window "Comp Air" to true
 unlock screen
end openCard

on mouseUpInPicture
 close window "comp air"
end mouseUpInPicture


••••• CARD No.5 Ventilation Exhaust
on openCard
 lock screen
 picture "Vent Exhaust", file, window, false
 set the globalrect of window "Vent Exhaust" to "32,32,630,430"
 set the scroll of window "Vent Exhaust" to "100,50"
 set the visible of window "Vent Exhaust" to true
 unlock screen
end openCard

on mouseUpInPicture
 close window "vent exhaust"
end mouseUpInPicture


••••• CARD No.6 Ventilation Supply
on openCard
 lock screen
 picture "Vent Supply", file, window, false
 set the globalrect of window "Vent Supply" to "32,32,630,430"
 set the scroll of window "Vent Supply" to "100,50"
 set the visible of window "Vent Supply" to true
 unlock screen
end openCard

on mouseUpInPicture
 close window "vent supply"
end mouseUpInPicture
```

```
••••• CARD No.7 Casualty Power
on openCard
 lock screen
 picture "Cas Power", file, window, false
 set the globalrect of window "Cas Power" to "32,32,630,430"
 set the scroll of window "Cas Power" to "100,50"
 set the visible of window "Cas Power" to true
 unlock screen
end openCard

on mouseUpInPicture
 close window "cas power"
end mouseUpInPicture


••••• CARD No.8 Main And Secondary Drains
on openCard
 lock screen
 picture "Main Drain", file, window, false
 set the globalrect of window "Main Drain" to "32,32,630,430"
 set the scroll of window "Main Drain" to "100,50"
 set the visible of window "Main Drain" to true
 unlock screen
end openCard

on mouseUpInPicture
 close window "main drain"
end mouseUpInPicture


••••• CARD No.9 Miscelaneous Drains
on openCard
 lock screen
 picture "Misc Drains", file, window, false
 set the globalrect of window "Misc Drains" to "32,32,630,430"
 set the scroll of window "Misc Drains" to "100,50"
 set the visible of window "Misc Drains" to true
 unlock screen
end openCard

on mouseUpInPicture
 close window "misc drains"
end mouseUpInPicture
```

```
••••• CARD No.10 FM Seg X & Y
on openCard
 lock screen
 picture "FM Seg X & Y", file, window, false
 set the globalrect of window "FM Seg X & Y" to "32,32,630,430"
 set the scroll of window "FM Seg X & Y" to "100,50"
 set the visible of window "FM Seg X & Y" to true
 unlock screen
end openCard

on mouseUpInPicture
 close window "fm seg x & Y"
 close window "fm seg z"
end mouseUpInPicture


••••• CARD No.10  BUTTON No.1  Zebra
on mouseUpInPicture
 close window "fm seg x & Y"
 lock screen
 push card
 go to card 11 of stack "dc diagrams"
 unlock screen
end mouseUpInPicture

on mouseUp
 close window "fm seg X & Y"
 push card
 go to card 11 of stack "dc diagrams"
end mouseUp


••••• CARD No.11 FM Seg Z
on openCard
 lock screen
 picture "FM Seg Z", file, window, false
 set the globalrect of window "FM Seg Z" to "32,32,630,430"
 set the scroll of window "FM Seg Z" to "100,50"
 set the visible of window "FM Seg Z" to true
 unlock screen
end openCard

on mouseUpInPicture
 close window "fm seg z"
end mouseUpInPicture
```

••••• CARD No.11 BUTTON No.1 FM SEG X & Y

```
on mouseUp
 close window "fm seg z"
 push card
 go to card "fm seg x & y"
end mouseUp
```

LIST OF REFERENCES

1. Giannotti, G. CDR. and Duffy, Kevin LT., *ARGOS: Design and Development of Object-oriented, Event-driven, Multimedia Database Technology in Support of the Paperless Ship*, Masters Thesis, Naval Postgraduate School, Monterey, CA., December 1988.

2. Apple Computer, Inc., *HyperCard User's Guide*, 1990.

3. Chief of Naval Operations OPNAV Instruction 3120.32B, *Standard Organization and Regulations of the U.S. Navy*, 26 September 1986.

4. Naval Sea Systems Command, *Naval Ships Technical Manual 9881*, revision 1989.

5. Naval Sea Systems Command, NAVSEA 9066/1, *Damage Control Book (FFG-7 Class)*, Reprint for SRA, 1989.

6. Awad, E.M., *Management Information Systems, Concepts, Structure, and Applications*, pp. 340-347, Benjamin/Cummings, 1988.

BIBLIOGRAPHY

Aker, S.Z., *The Macintosh Bible*, 3d ed., Goldstein & Blair, 1991.

Goodman, Danny, *The Complete HyperCard 2.0 Handbook*, 2d ed., Bantam Books, 1990.

Goodman, Paul, *Power User's Hypertalk Handbook*, Windcrest, 1989.

Kroenke, D.M., and Dolan, K.A., *Database Processing, Fundamentals, Design, Implementation*, 3d ed., Macmillan publishing company, 1988.

Powers, M.J., Cheney, P.H., and Crow, G., *Structured Systems Development, Analysis, Design, Implementation*, 2d ed., Boyd & Fraser, 1990.

Waite, M., Prata, S., Jones, T., *The Waite Groups Hypertalk Bible*, Hayden Books, 1989.

Whitten, J.L., Bentley, L.D., and Barlow, V.M., *Systems Analysis & Design Methods*, 2d ed., Irwin, 1989.

Winkler, Dan, and Kamins, Scot, *Hypertalk 2.0: The Book*, Bantam Books, 1990.